

## Para saber más: Java Record

Lanzado oficialmente en Java 16, pero disponible experimentalmente desde Java 14. **Record** es un recurso que le permite representar una clase inmutable, que contiene solo atributos, constructor y métodos de lectura, de una manera muy simple y ágil.

Este tipo de clase encaja perfectamente para representar clases DTO, ya que su objetivo es únicamente representar datos que serán recibidos o devueltos por la API, sin ningún tipo de comportamiento.

Para crear una clase DTO inmutable, sin la utilización de Record, era necesario escribir mucho código. Veamos un ejemplo de una clase DTO que representa un teléfono:

```
public final class Telefono {  
  
    private final String ddd;  
    private final String numero;  
  
    public Telefono(String ddd, String numero) {  
        this.ddd = ddd;  
        this.numero = numero;  
    }  
  
    @Override  
    public int hashCode() {  
        return Objects.hash(ddd, numero);  
    }  
}
```

@Override

```
public boolean equals(Object obj) {  
    if (this == obj) {  
        return true;  
    } else if (!(obj instanceof Telefono)) {  
        return false;  
    } else {  
        Telefono other = (Telefono) obj;  
        return Objects.equals(ddd, other.ddd)  
            && Objects.equals(numero, other.numero);  
    }  
}  
  
public String getDdd() {  
    return this.ddd;  
}  
  
public String getNumero() {  
    return this.numero;  
}  
}
```

COPIA EL CÓDIGO

Ahora, con Record todo ese código se puede resumir en una sola línea:

```
public record Telefono(String ddd, String numero){}
```

COPIA EL CÓDIGO

Mucho más simple, ¿no?

Bajo el capó, Java transformará este registro en una clase inmutable, muy similar al código que se muestra arriba.

Se pueden encontrar más detalles sobre esta función en la [documentación oficial \(https://docs.oracle.com/en/java/javase/16/language/records.html\)](https://docs.oracle.com/en/java/javase/16/language/records.html).